Iterative Rounding for Multi-Objective Optimization Problems

Fabrizio Grandoni[1], R. Ravi[2], and Mohit Singh[3]

[1] Department of Computer Science, Systems and Production, University of Rome Tor Vergata
grandoni@disp.uniroma2.it

[2] Tepper School of Business, Carnegie Mellon University, Pittsburgh, USA
Supported in part by NSF grant CCF-0728841
ravi@cmu.edu

[3] Microsoft Research, New England, Cambridge, USA
mohsingh@microsoft.com

Abstract. In this paper we show that *iterative rounding* is a powerful and flexible tool in the design of approximation algorithms for multi-objective optimization problems. We illustrate that by considering the multi-objective versions of three basic optimization problems: spanning tree, matroid basis and matching in bipartite graphs. Here, besides the standard weight function, we are given k length functions with corresponding budgets. The goal is finding a feasible solution of maximum weight and such that, for all i, the ith length of the solution does not exceed the ith budget. For these problems we present polynomial-time approximation schemes that, for any constant $\epsilon > 0$ and $k \geq 1$, compute a solution violating each budget constraint at most by a factor $(1+\epsilon)$. The weight of the solution is optimal for the first two problems, and $(1-\epsilon)$-approximate for the last one.

1 Introduction

Most real-life optimization problems involve finding a feasible solution trading off many mutually conflicting goals. This is a rich area of study in Operations Research, Economics and Computer Science in the broad area of *Multi-objective Optimization* [10,14,26]. A variety of approaches have been employed to formulate such problems including Goal Programming [4], Pareto-Optimality [9], and Multi-objective Approximation Algorithms [26]. We adopt the latter approach and cast one of the goals as the objective function, and the others as *budget constraints.* More precisely, we are given a (finite) set $\mathcal{F}$ of feasible solutions for the problem; we are also given a weight function $\mathrm{w} : \mathcal{F} \rightarrow \mathbb{R}_+$ and a set of k length functions $\ell^i : \mathcal{F} \rightarrow \mathbb{R}_+$, $1 \leq i \leq k$, that assign a weight $\mathrm{w}(S)$ and k lengths $\ell^i(S)$, $1 \leq i \leq k$, to every feasible solution $S \in \mathcal{F}$. For each length function ℓ^i, we are also given a non-negative budget $\mathrm{L}_i \in \mathbb{R}_+$. The *multi-objective optimization problem* can then be formulated as follows[1].

[1] With a slight notation abuse, we will use OPT also to denote the actual optimal solution (besides its weight).

$$\text{OPT} := \text{maximize } \mathrm{w}(S) \text{ subject to } S \in \mathcal{F},\ \ell^i(S) \leq \mathrm{L}_i,\ 1 \leq i \leq k. \quad (1)$$

In this paper we study the multi-objective version of three fundamental maximization problems, namely spanning tree, matroid basis, and matching in bipartite graphs.

1. In the MULTI-OBJECTIVE SPANNING TREE problem, we are given an n-node undirected graph $G = (V, E)$ with edge weights $\mathrm{w} : E \to \mathbb{R}_+$, k edge lengths $\ell^i : E \to \mathbb{R}_+$, $1 \leq i \leq k$, and positive budgets $\mathrm{L}_1, \ldots, \mathrm{L}_k$. The set of all feasible solutions $\mathcal{F}$ is given by the spanning trees of G. Define the weight of $T \in \mathcal{F}$ as $\mathrm{w}(T) := \sum_{e \in T} \mathrm{w}(e)$, and its ith-length as $\ell^i(T) := \sum_{e \in T} \ell^i(e)$. The goal is finding $T \in \mathcal{F}$ of maximum weight $\mathrm{w}(T)$ such that $\ell^i(T) \leq \mathrm{L}_i$ for each $1 \leq i \leq k$.

2. The MULTI-OBJECTIVE BIPARTITE MATCHING problem is defined analogously. Here the goal is finding a matching M in a bipartite graph of maximum-weight $\mathrm{w}(M)$ such that $\ell^i(M) \leq \mathrm{L}_i$ for all $1 \leq i \leq k$.

3. In the MULTI-OBJECTIVE MATROID BASIS problem, we are given a matroid $\mathcal{M} = (E, \mathcal{E})$, $\mathcal{E} \subseteq 2^E$, on the ground set E, $m = |E|$ (for basic definitions and results on matroids, see e.g. [27]). Moreover, we are given element weights $\mathrm{w} : E \to \mathbb{R}_+$, element lengths $\ell^i : E \to \mathbb{R}_+$ and budgets $\mathrm{L}_i \in \mathbb{R}_+$, $1 \leq i \leq k$. The set of all feasible solutions $\mathcal{F}$ is given by then bases of $\mathcal{M}$. The weight of a basis $B \in \mathcal{E}$ is defined as $\mathrm{w}(B) := \sum_{e \in B} \mathrm{w}(e)$, while its ith-length is $\ell^i(B) := \sum_{e \in B} \ell^i(e)$. The goal is computing a basis $X \in \mathcal{F}$ of maximum weight satisfying all the budget constraints. This naturally generalizes the multi-objective spanning tree problem which results when we consider a graphic matroid.

All three problems are polynomial-time solvable in their unbudgeted version ($k = 0$), but become NP-hard [1,6] even for a single budget constraint ($k = 1$).

Our Results. We give a PTAS for multi-objective spanning trees and generalize it to multi-objective matroid basis and also give a PTAS for multi-objective matchings in bipartite graphs. Our results however require that the number of budget constraints k is fixed.

Theorem 1. *For any $\epsilon > 0$, there exists an algorithm for* MULTI-OBJECTIVE SPANNING TREE *with $k \geq 1$ budget constraints which returns a spanning tree T of optimal weight and $\ell^i(M) \leq (1+\epsilon)\mathrm{L}_i$ for each $1 \leq i \leq k$. The running time of the algorithm is $O(n^{O(k^2/\epsilon)})$.*

Theorem 1, proved in Section 2, generalizes the result of Ravi and Goemans [25] who gave the same guarantees for the special case of a single budget constraint($k = 1$), and improves on the (much more involved) algorithm of Papadimitriou and Yannakakis [22] which returns a suboptimal ($(1-\epsilon)$-approximate) solution with a similar (i.e., $1+\epsilon$) violation of the budget constraints. The latter result of [22] also holds for our case of many different but fixed number of objectives, and even in this case, we improve on the approximation factor in the main objective (with the same violation in the budgets).

Theorem 2. *For any $\epsilon > 0$, there exists an algorithm for* Multi-Objective matroid basis *with $k \geq 1$ budget constraints which returns a basis B of optimal weight and $\ell^i(B) \leq (1+\epsilon)\mathrm{L}_i$ for each $1 \leq i \leq k$. The running time of the algorithm is $O(m^{O(k^2/\epsilon)})$.*

Theorem 2 is discussed in Section 2.1, and generalizes a similar result for the $k = 1$ case as above in [25].

Theorem 3. *. For any $\epsilon > 0$, there exists a deterministic algorithm for* Multi-Objective Bipartite Matching *with $k \geq 1$ budget constraints which returns a matching M of weight $\mathrm{w}(M) \geq (1-\epsilon)\mathrm{OPT}$ and length $\ell^i(M) \leq (1+\epsilon)\mathrm{L}_i$ for each $1 \leq i \leq k$. The running time of the algorithm is $O(n^{O(k^2\sqrt{k \log k}/\epsilon^2)})$.*

Theorem 3 is proved in Section 3. A similar approximation guarantee was known earlier via the work of [22]. However, their result implies a fully polynomial RNC scheme rather than a PTAS, and thus Theorem 3 provides the first deterministic approximation scheme for multi-objective bipartite matching. A PTAS, based on a completely different approach, was known earlier only for the case of one budget constraint, i.e. $k = 1$ [6].

Our Techniques. Perhaps even more importantly than our specific results, our main contribution is to demonstrate that the general framework of *iterative techniques* can be used to obtain approximation algorithms for various multi-objective optimization problems. This technique was introduced by Jain [15] for approximating survivable network design problems. The basic idea in iterative *rounding* for covering problems is as follows: Consider the optimal (fractional) vertex (or extreme point or basic feasible) solution to a linear programming relaxation to the problem, and show that there is a variable with high fractional value (e.g. at least 0.5) which can be rounded up to an integer without losing too much (e.g. 2) in the approximation. The method includes this rounded variable in the integral solution and iterates. Since the basis iterative rounding loses a constant factor in approximation, we refine the method by replacing the rounding step by the following: relax (remove) a constraint that can be ignored without losing too much in the feasibility and iterate on the residual problem. The resulting iterative *relaxation* method has been very successful for approximating degree-constrained network design problems [16,17,29].

We now outline how the iterative technique is applied to our problems. The algorithm for multi-objective spanning tree is rather simple; a vertex solution for the natural LP relaxation of the problem is already sparse: it has about k edges more than a spanning tree in its support due to the well-known laminarity of an independent set of tight spanning tree constraints [27]. We remove all edges corresponding to variables of value zero, relax (remove) all the budget constraints, and solve optimally the residual problem (which is a standard spanning tree problem). A preliminary guessing phase ensures that the k edges not used in the tree do not add much to the approximation bound for any of the budgets. This approach also gives a very simple proof of the earlier result for the case $k = 1$ [25]. An identical approach works also for the more general MULTI-OBJECTIVE MATROID BASIS problem.

Our algorithm for MULTI-OBJECTIVE BIPARTITE MATCHING is more involved: after an initial preprocessing phase, where the algorithm removes all edges with large weight and large length, there is a decomposition phase. In that phase, we run an iterative relaxation algorithm which uses the optimal solution of the natural LP formulation to obtain a modified LP solution. The iterative algorithm ensures that the support of the modified solution is a collection of $h \leq k$ vertex disjoint paths. Moreover, each of these paths has small weight and length. In the final combination phase, we combine the solutions on these paths to return one feasible matching. Each path can be decomposed in two matchings. The algorithm picks one matching from each of the paths. While the algorithm is a brute force enumeration over all choices (which are $2^h \leq 2^k$ in number), a probabilistic argument is used to show that there exists a choice of a matching from each path which provides a solution with the desired guarantee.

Related Work. Multi-objective optimization has been studied extensively in Operations Research, Microeconomics and Computer Science. We refer the reader to more general sources [4,9,10,14], and restrict our attention to work which closely relates to our problems. There are many examples of single-budget versions of polynomial-time solvable optimization problems addressed in the literature. In the *constrained shortest path problem* the goal is finding a minimum-weight path in a directed graph between two nodes s and t such that the length of the path does not exceed a budget L [5]. In the *constrained minimum arborescence problem* we are given a directed graphs with edge weights and lengths. The aim is computing an *arborescence* of minimum weight whose length is below the input budget [13]. Previous work on budgeted optimization problems also includes results on budgeted scheduling [18,28] and bicriteria results for several budgeted network design problems [19].

Jain [15] introduced the iterative rounding framework and applied it to approximating general network design problems. Subsequently, it was applied to various other network design problems [8,12,20]. The iterative relaxation technique has recently been successfully applied to degree constrained network design problems [2,16,17,29].

There are few general tools for designing approximation algorithms for budgeted problems. One is the Lagrangian relaxation method. The basic idea is relaxing the budget constraint, and lifting it into the objective function weighting it by a Lagrangian multiplier. Solving the relaxed problem, one obtains two or more optimal solutions, which are then patched together to get a good solution for the original problem. Demonstrating this method, Goemans and Ravi [25] gave the first PTAS for MULTI-OBJECTIVE SPANNING TREE with a single budget constraint. Using the same approach, but a more involving patching step, Berger, Bonifaci, Grandoni, and Schäfer [6] obtained a PTAS for the single-budget version of the matching problem. This approach does not seem to generalize to the case of multiple budget constraints.

A second general tool, due to Papadimitrou and Yannakakis [22], is based on the construction of succinct approximation of Pareto curves. In order to efficiently construct such ϵ-approximate Pareto curves, a sufficient condition is the existence of a pseudo-polynomial-time algorithm for the *exact* version of the problem considered. The task in the exact version of the problem is to return a feasible solution of *exactly* some pre-specified value. The existence of such pseudo-polynomial-time algorithm for the spanning tree problem [3] implies a polynomial-time algorithm which returns a $(1-\epsilon)$-approximate solution violating all the budget constraints by a factor of $(1+\epsilon)$ for the corresponding multi-objective version. Unfortunately, it is not known whether such an algorithm exists for matchings in bipartite graphs, while the famous randomized algorithm of Mulmuley, Vazirani and Vazirani [21] can be used to obtain a polynomial-time randomized approximation scheme for MULTI-OBJECTIVE BIPARTITE MATCHING[2]. Their method, however, only approximates the objective while our algorithm matches the value of the objective function with the optimal for two out of the three problems addressed here, while for the third we obtain a deterministic rather than an RNC algorithm.

A third approach is based on parametric search and is advocated in [19]; their results imply that a ρ-approximation algorithm for the single objective problem gives a $(k \cdot \rho)$-approximation for *each* of the budget violations as well as for the objective in the corresponding k-objective problem. This only gives a much weaker k-approximation for each objective for the problems considered here.

Other general tools for multi-objective problems such as Matching-Based Augmentation [24] advocates building the solution iteratively using one (path) matching at a time controlling the various objectives, and Randomized Rounding of fraction LP solutions while bounding all objectives simultaneously [7,23]. While these techniques are useful in handling more than one type of objective, their performance ratios tend to be in the higher logarithmic range.

In the context of these methods, our paper shows that iterative rounding is a powerful and flexible tool for approximating multi-objective optimization problems giving even better results than all of the above methods. This was already the case for degree-constrained spanning trees and survivable network design problems [16,29] and directed network design problems [2], and our results extend these to some more multi-objective problems.

2 Multi-Objective Spanning Tree and Matroid Basis

We formulate the following linear programming relaxation for MULTI-OBJECTIVE SPANNING TREE which is a standard extension of the linear program for the maximum spanning tree problem. There is a variable x_e for each edge $e \in E$. For a subset $F \subseteq E$ of edges, we denote $x(F) = \sum_{e \in F} x_e$ and for a subset $S \subseteq V$, we denote $E(S) = \{e : |e \cap S| = 2\}$ to be the set of edges with both endpoints in S.

[2] The same algorithm works for general graphs also.

$$
\begin{array}{llll}
\text{(LP-ST)} & \text{maximize} & \displaystyle\sum_{e\in E} \mathrm{w}(e)\, x_e & \\
& \text{subject to} & x(E(V)) = |V| - 1, & \\
& & x(E(S)) \le |S| - 1, & \forall\, S \subset V \\
& & \displaystyle\sum_{e\in E} \ell^i(e) x_e \le \mathrm{L}_i, & \forall\, 1 \le i \le k \\
& & x_e \ge 0, & \forall\, e \in E.
\end{array}
$$

The following characterization of any vertex solution of (LP-ST) follows directly from the uncrossing technique (see [27]).

Lemma 1. *Let x be a vertex solution of the linear program (LP-ST) such that $x_e > 0$ for each edge e and let $\mathcal{T} = \{S \subseteq V : x(E(S)) = |S| - 1\}$ be the set of all tight subset constraints. Then there exists a laminar family $\mathcal{L} \subseteq \mathcal{T}$ and a subset $J \subseteq \{1 \le i \le k : \sum_{e\in E} \ell^i(e) x_e = \mathrm{L}_i\}$ of tight length constraints such that*

1. *The vectors $\{\chi(E(S)) : S \in \mathcal{L}\}$ are linearly independent.*
2. *span($\mathcal{L}$)=span($\mathcal{T}$)*
3. *$|\mathcal{L}| + |J| = |E|$*

Algorithm for Multi-Objective Spanning Tree
1. Guess all edges in the optimal solution such that $\ell^i(e) \ge \frac{\epsilon}{k}\mathrm{L}_i$. Include these edges in the solution and contract them. Delete all other edges with $\ell^i(e) \ge \frac{\epsilon}{k}\mathrm{L}_i$ from G. Update L_i.
2. Find a vertex solution x of (LP-ST) for the residual problem and remove every edge e with $x_e = 0$.
3. Pick any maximum-weight spanning tree in the support.

The algorithm for MULTI-OBJECTIVE SPANNING TREE above proceeds in two phases. The first phase is the pruning step which we describe below. Observe that no feasible solution can include an edge whose ith-length is more than L_i. We extend this step further and *guess* all edges in the solution whose ith-length is at most $\frac{\epsilon}{k}\mathrm{L}_i$. For any i there can be at most $\frac{k}{\epsilon}$ such edges in the optimal solution. Hence, trying all such possibilities for inclusion in a partial initial solution takes time $O(m^{k/\epsilon})$ where m is the number of edges in G. There are k length function to try which amounts to the total number of choices being at most $O(m^{k^2/\epsilon})$. After guessing these edges correctly, we throw away all other edges which have ℓ^i length more than $\epsilon\mathrm{L}_i$ and contract the guessed edges in the optimal solution. Clearly, the rest of the edges in the optimal solution form a spanning tree in the contracted graph. Also, now we have an instance where $\ell^i(e) \le \frac{\epsilon}{k}\mathrm{L}_i$ for each e and i. We also update the bound L_i by subtracting the lengths of the selected edges. Let L'_i denote the residual bounds. We solve the linear program (LP-ST) with updated bounds L'_i. Step (3) can be interpreted as removing all the k constraints bounding the length under the length functions $l^1 \ldots, l^k$. Removing these constraints gives us the linear program for the spanning tree problem which is integral and its optimal solution is a maximum weight spanning tree.

Proof. (Theorem 1) First observe that the support of (LP-ST) on a graph with n vertices has at most $n+k-1$ edges. In fact, from Lemma 1, we have $|E| = |\mathcal{L}|+|J|$. But $|\mathcal{L}| \leq n-1$ since $\mathcal{L}$ is a laminar family without singletons and $|J| \leq k$ proving the claim.

Observe that the weight of the tree returned by the algorithm is at most the weight of the LP-solution and hence is optimal for the correct guess of *heavy* edges. Now, we show that the ith-length is at most $L'_i + \epsilon \mathrm{L}_i$. Observe that any tree must contain $n-1$ edges out of the $n+k-1$ edges in the support. Hence, the maximum ith-length tree has length no more than $k \cdot \frac{\epsilon}{k}\mathrm{L}_i = \epsilon \mathrm{L}_i$ more than the minimum ith-length tree. In turn, the tree of minimum ith-length has ith-length no larger than the ith-length of the optimal fractional solution, which is at most L'_i by feasibility. Altogether, the maximum ith-length of the solution returned is no more than $\mathrm{L}'_i + \epsilon \mathrm{L}_i$. Adding the length of edges guessed in the first step we obtain that the tree returned by the algorithm has ith-length at most $\mathrm{L}'_i + \epsilon \mathrm{L}_i + \mathrm{L}_i - \mathrm{L}'_i = (1+\epsilon)\mathrm{L}_i$.

2.1 Multi-Objective Matroid Basis

The results on Multi-objective spanning tree can be naturally generalized to the case of Multi-objective matroid basis. Consider the following linear programming relaxation (LP-MB) for the problem. There is a variable x_e for each element $e \in E$. For any subset $S \subseteq E$, we denote $x(S) = \sum_{e \in S} x_e$. Here r denotes the rank function of the matroid $\mathcal{M}$.

$$
\begin{array}{lll}
\text{(LP-MB)} \quad \text{maximize} & \displaystyle\sum_{e \in E} \mathrm{w}(e)\, x_e & \\
\text{subject to} & x(E) = r(E), & \\
& x(S) \leq r(S), & \forall\, S \subseteq E \\
& \displaystyle\sum_{e \in E} \ell^i(e) x_e \leq \mathrm{L}_i, & \forall\, 1 \leq i \leq k \\
& x_e \geq 0, & \forall\, e \in E.
\end{array}
$$

The polynomial time solvability of the linear program (LP-MB) follows from the polynomial time separation of the rank constraints [11]. Our algorithm for Multi-objective matroid basis is described below. Its analysis follows along the same line as in the case of Multi-objective spanning tree and is omitted due to space restrictions.

Algorithm for Multi-Objective Matroid Basis

1. Guess all elements in the optimal solution such that $\ell^i(e) \geq \frac{\epsilon}{k}\mathrm{L}_i$. Include all such elements in the solution and update the matroid by contracting these elements in the matroid. Delete all other heavy elements e with $\ell^i(e) \geq \frac{\epsilon}{k}\mathrm{L}_i$ for any i from $\mathcal{M}$. Update L_i.
2. Find a vertex solution x of (LP-MB) for the residual problem and remove every element e with $x_e = 0$.
3. Pick any maximum weight basis in the support.

3 Multi-Objective Bipartite Matching

In this section we present a polynomial-time approximation scheme for Multi-Objective Bipartite Matching and prove Theorem 3.

We formulate the following linear programming relaxation (LP-BM) for the problem. We use $\delta(v)$ to denote the set of edges incident to $v \in V$.

$$\begin{array}{llll}
\text{(LP-BM)} & \text{maximize} & \sum_{e\in E} \mathrm{w}(e)\, x_e & \\
& \text{subject to} & \sum_{e\in\delta(v)} x_e \le 1, & \forall v \in V \\
& & \sum_{e\in E} \ell^i(e) x_e \le \mathrm{L}_i, & \forall 1 \le i \le k \\
& & x_e \ge 0, & \forall e \in E.
\end{array}$$

Algorithm for Multi-Objective Bipartite Matching

Preprocessing

(a) Let $\delta = \epsilon^2 / 36k\sqrt{2k\ln(k+2)}$. Guess all the edges e in OPT such that $\mathrm{w}(e) \ge \delta\, OPT$ or $\ell^i(e) \ge \delta\, \mathrm{L}_i$ for some i, and add them to the solution. Reduce the problem consequently.

Decomposition

(b) Compute the optimal fractional vertex solution x^b to LP-BM for the reduced problem. As long as there is an integral variable, reduce the problem appropriately and iterate.

(c) Remove all the nodes of degree zero and of degree at least 3, and all the edges incident to the removed nodes. Compute an optimal fractional vertex solution x^c to the problem LP-BM in the remaining graph. As long as there is an integral variable, reduce the problem appropriately and iterate. Finally, remove one edge from each remaining cycle.

(d) Compute an optimal fractional vertex solution x^d to the problem LP-BM in the remaining graph. As long as there is an integral variable, reduce the problem appropriately and iterate.

(e) Let $\gamma = \epsilon / 2\sqrt{2k\ln(k+2)}$. As long as there is a path $P = (e_1, e_2, \ldots, e_t)$ induced by x^d such that $\mathrm{w}(P) > \gamma\, \mathrm{w}(x^d)$ or $\ell^i(P) > \gamma\, \ell^i(x^d)$ for some i, find a minimal subpath $P' = (e_1, e_2, \ldots, e_{t'})$ of P satisfying the condition above and remove $e_{t'}$ from the graph.

(f) Compute an optimal fractional vertex solution x^f to the problem LP-BM in the remaining graph. As long as there is an integral variable, reduce the problem appropriately and iterate.

(g) Let $P_1, P_2, \ldots, P_q$ be the set of paths induced by x^f. Return the subpaths $S_1, S_2, \ldots, S_h$ formed after deleting the internal nodes whose matching constraints are not tight with respect to x^f. Return the solution x^g which is x^f induced on the edges in S_i for each $1 \le i \le h$.

Combination

(h) Let M_j and $\bar{M}_j$ be the two matchings partitioning S_j. Return the matching M' satisfying the following properties: (i) For each S_j, $M' \cap S_j \in \{M_j, \bar{M}_j\}$; (ii) $\mathrm{w}(M') \ge (1-\epsilon/2)\mathrm{w}(x^g)$ and $\ell^i(M') \le (1+\epsilon/2)\ell^i(x^g)$ for all i.

The algorithm for Multi-Objective Bipartite Matching above works in three phases.

In the *Preprocessing Phase*, the algorithm guesses all the edges in OPT of weight at least $\delta\, OPT$ or ith-length at least $\delta \mathrm{L}_i$ for some i. Here δ is a proper function of ϵ and k. This guessing can be performed in time polynomial in n (but exponential in δ). The algorithm then includes all the guessed edges in the solution, and deletes the remaining heavy edges. It also reduces the L_i's accordingly. After this phase $\mathrm{w}(e) \leq \delta\, OPT$ and $\ell^i(e) \leq \delta L_i$ for each edge e.

In the *Decomposition Phase* our algorithm computes over a series of pruning and iterative steps, a solution to the multi-objective matching problem on a reduced graph that is eventually a collection of paths. In Step (c), we discard nodes of degree 0 or of degree 3 or higher so as to leave only paths and cycles; Finally, one edge from each cycle is removed in this step. In Step (e), we further break each path into subpaths of bounded total weight and length. This pruning is useful in the later Combination Phase when we choose one of the two matchings in each path: the bounded difference ensures that one such combination is near optimal. The use of vertex solutions in all the residual problems ensures that the total number of edges thrown away in all the above stages is roughly of the order of the extra budget constraints in the problem which is $O(k/\gamma)$ for a parameter $\gamma \simeq O(\epsilon/\sqrt{k})$. Finally, we output a feasible fractional vertex solution x^g to the LP with the following properties.

(1) The support of x^g is a collection of vertex disjoint paths $S_1, \ldots, S_h$ where $h \leq k$.
(2) x^g is a $(1 + \epsilon/4)$-approximate solution.
(3) For each S_i, the degree constraints of the vertices of S_i are tight except for its endpoints.
(4) For each S_i, $w \cdot x^g(S_i) \leq \gamma OPT$ and $\ell_i \cdot x^g(S_j) \leq \gamma L_i$ for each $1 \leq i \leq k$ and $1 \leq j \leq h$ where $\gamma = \epsilon/2\sqrt{2k \ln(k+2)}$.

In the final *Combination Phase*, the paths $S_1, \ldots, S_h$ are used to compute an approximate feasible (integral) solution. The algorithm enumerates over all the 2^h matchings which are obtained by taking, for each S_i, one of the two matchings which partition S_i. This enumeration takes polynomial time since $h \leq k = O(1)$. A probabilistic argument is used to show that one of these matchings satisfies the claimed approximation guarantee of the algorithm.

Analysis. We now analyze the three phases of the algorithm, bounding the corresponding approximation guarantee and running time. Consider first the *Preprocessing Phase*. In order to implement Step (a), we have to consider all the possible choices, and run the algorithm for each choice. Observe that there are at most $(k+1)/\delta$ such heavy edges in the optimal solution, and hence the number of possibilities is $O(m^{(k+1)/\delta}) = O(m^{O(k^2\sqrt{k \log k}/\epsilon^2)})$. The algorithm generates a different subproblem for each possible guess of the edges. In the following we will focus on the run of the algorithm where the guessed edges correspond to an optimal solution to the multi-objective problem.

Consider now the *Decomposition Phase*. We prove that the output of this phase satisfies the four properties stated above. Observe that by construction the algorithm returns a collection of edge disjoint paths whose interior vertices have tight degree constraints. Properties (3) and (4) follow by construction. We now argue that the number of paths is bounded by k, proving Property (1).

Lemma 2. *The number h of subpaths in Step (g) is upper bounded by k.*

Proof. Consider the solution x^f. The number of variables $|E| = \sum_{i=1}^{q} |P_i|$ is upper bounded by the number of tight constraints. Let q' be the number of internal nodes whose matching constraint is not tight in x^f. Note that the matching constraints at the endpoints of each path are not tight. Hence the number of tight constraints is at most $\sum_{i=1}^{q}(|P_i|-1)-q'+k = |E|-q-q'+k \geq |E|$, from which $q+q' \leq k$. Observe that, by definition, the number h of subpaths is exactly $q+q'$ (we start with q subpaths, and create a new subpath for each internal node whose matching constraint is not tight). The claim follows.

Clearly, solution x^g satisfies all the constraints. We next argue that the weight of x^g is nearly optimal. In Steps (c), (e) and (g) we remove a subset of edges whose optimal fractional value is larger than zero in the step considered. In the following lemma, whose proof is omitted for lack of space, we bound the number of edges removed. Due to the Preprocessing Phase, the weight of these edges is negligible, which implies that the consequent worsening of the approximation factor is sufficiently small. This proves Property (2).

Lemma 3. *The algorithm removes at most $7k$, $(k+1)/\gamma$, and $2k$ edges in Steps (c), (e), and (g), respectively.*

Each of the steps (b) to (g) is run polynomially many times and takes polynomial time. Hence the overall running time of the Decomposition Phase is polynomial.

Consider eventually the *Combination Phase.* As described earlier, the running time of this phase is bounded by $O(2^k n^{O(1)})$. The following lemma, which is the heart of our analysis, shows that a subset M' satisfying Properties (i) and (ii) always exists. Henceforth the algorithm always returns a solution. Although we use a randomized argument to prove the lemma, the algorithm is completely deterministic and enumerates over all solutions. Recall that M_j and $\bar{M}_j$ are the two matchings which partition subpath S_j.

Lemma 4. *In Step (h) there is always a set of edges M' satisfying Properties (i) and (ii).*

Proof. Consider the following packing problem

$$\begin{array}{lll}
(PACK) & \text{maximize} & \displaystyle\sum_{j=1}^{h}(y_j\,\mathrm{w}(M_j) + (1-y_j)\,\mathrm{w}(\bar{M}_j)) \\
& \text{subject to} & \displaystyle\sum_{j=1}^{h}(y_j\,\ell^i(M_j) + (1-y_j)\,\ell^i(\bar{M}_j)) \leq \mathrm{L}_i, \qquad \forall\, 1 \leq i \leq k \\
& & y_j \in \{0,1\}, \qquad \forall\, 1 \leq j \leq h.
\end{array}$$

We can interpret the variables y_j in the following way: $M' \cap S_j = M_j$ if $y_j = 1$, and $M' \cap S_j = \bar{M}_j$ otherwise. Given a (possibly fractional and infeasible) solution y to PACK, we use $\mathrm{w}(y)$ and $\ell^i(y)$ as shortcuts for $\sum_{j=1}^{h}(y_j\,\mathrm{w}(M_j) + (1-y_j)\,\mathrm{w}(\bar{M}_j))$ and $\sum_{j=1}^{h}(y_j\,\ell^i(M_j) + (1-y_j)\,\ell^i(\bar{M}_j))$, respectively.

Observe that x^g induces a feasible fractional solution y^g to the linear relaxation of PACK. In fact, consider each subpath S_j. By definition, each matching constraint at an internal node of S_j is tight. This implies that all the edges e of M_j (resp., $\bar{M}_j$) have the same value $x^g_e =: y^g$ (resp., $x^g_e =: 1 - y^g$). Thus, we have $\mathrm{w}(y^g) = \mathrm{w}(x^g)$. Now, we construct a (near) feasible integral solution y' to PACK which satisfies (i) and (ii). Independently, for each path S_i, select M_i with probability y^g_i and $\bar{M}_i$ with probability $1 - y^g_i$. Note that $E[\mathrm{w}(y')] = \mathrm{w}(y^g)$ and $E[\ell^i(y')] = \ell^i(y^g) \leq \mathrm{L}_i$ for all i.

By Step (e), switching one variable of y' from 1 to 0 or vice versa can change the cost and ith-length of y' at most by $\gamma\,\mathrm{w}(x^g)$ and $\gamma\,\ell^i(x^g)$, respectively. The proof of the lemma now follows directly from the following proposition, which derives from Chernoff's bounds.

Proposition 1. *With positive probability, $w(y') \geq (1-\epsilon/2)w(x^g)$ and $l^i(y') \leq (1+\epsilon/2)l^i(x^g)$ for all i.*

Proof. (Theorem 3) It is easy to see that the solution returned is a matching. Moreover a solution is always returned by Lemma 4. The approximation guarantee of the algorithm follow from the properties of the Decomposition step and Lemma 4. The running time of each step is polynomial (for fixed k and ϵ) thus proving Theorem 3.

Acknowledgement. We thank an anonymous referee for suggesting improvements in Section 3 in an earlier draft of the paper.

References

1. Aggarwal, V., Aneja, Y.P., Nair, K.P.K.: Minimal spanning tree subject to a side constraint. Computers & Operations Research 9, 287–296 (1982)
2. Bansal, N., Khandekar, R., Nagarajan, V.: Additive Guarantees for Degree Bounded Directed Network Design. In: STOC, pp. 769–778 (2008)
3. Barahona, F., Pulleyblank, W.R.: Exact arborescences, matchings and cycles. Discrete Applied Mathematics 16(2), 91–99 (1987)
4. Barichard, V., Ehrgott, M., Gandibleux, X., T'Kindt, V. (eds.): Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications. Lecture Notes in Economics and Mathematical Systems, vol. 618. Springer, Heidelberg (2009)
5. Beasley, J.E., Christofides, N.: An algorithm for the resource constrained shortest path problem. Networks 19, 379–394 (1989)
6. Berger, A., Bonifaci, V., Grandoni, F., Schäfer, G.: Budgeted matching and budgeted matroid intersection via the gasoline puzzle. In: Lodi, A., Panconesi, A., Rinaldi, G. (eds.) IPCO 2008. LNCS, vol. 5035, pp. 273–287. Springer, Heidelberg (2008)
7. Bilò, V., Goyal, V., Ravi, R., Singh, M.: On the Crossing Spanning Tree Problem. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) RANDOM 2004 and APPROX 2004. LNCS, vol. 3122, pp. 51–64. Springer, Heidelberg (2004)

8. Cheriyan, J., Vempala, S., Vetta, A.: Network design via iterative rounding of setpair relaxations. Combinatorica 26(3), 255–275 (2006)
9. Chinchuluun, A., Pardalos, P.M., Migdalas, A., Pitsoulis, L. (eds.): Pareto Optimality, Game Theory and Equilibria. Optimization and Its Applications, vol. 17 (2008)
10. Climacao, J.: Multicriteria Analysis. Springer, Heidelberg (1997)
11. Cunningham, W.H.: Testing Membership in Matroid Polyhedra. Journal of Combinatorial Theory B 36(2), 161–188 (1984)
12. Fleischer, L., Jain, K., Williamson, D.P.: Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. Journal of Computer and System Sciences 72(5), 838–867 (2006)
13. Guignard, M., Rosenwein, M.B.: An application of Lagrangean decomposition to the resource-constrained minimum weighted arborescence problem. Networks 20, 345–359 (1990)
14. Hartley, R.: Survey of Algorithms for Vector Optimization Problems. In: French, S., Hartley, R., Thomas, L.C., White, D.J. (eds.) Multiobjective Decision Making, pp. 1–34. Academic Press, London (1983)
15. Jain, K.: A factor 2 approximation algorithm for the generalized steiner network problem. Combinatorica 21, 39–60 (2001)
16. Lau, L.C., Naor, S., Salavatipour, M., Singh, M.: Survivable network design with degree or order constraints. In: STOC, pp. 651–660 (2007)
17. Lau, L.C., Singh, M.: Additive approximation for bounded degree survivable network design. In: STOC, pp. 759–768 (2008)
18. Levin, A., Woeginger, G.J.: The constrained minimum weight sum of job completion times. Mathematical Programming 108, 115–126 (2006)
19. Marathe, M.V., Ravi, R., Sundaram, R., Ravi, S.S., Rosenkrantz, D.J., Hunt III, H.B.: Bicriteria network design problems. In: Fülöp, Z., Gecseg, F. (eds.) ICALP 1995. LNCS, vol. 944, pp. 487–498. Springer, Heidelberg (1995)
20. Melkonian, V., Tardos, E.: Algorithms for a Network Design Problem with Crossing Supermodular Demands. Networks 43, 4 (2004)
21. Mulmuley, K., Vazirani, U., Vazirani, V.: Matching is as Easy as Matrix Inversion. Combinatorica 7(1), 101–104 (1987)
22. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of Web sources. In: FOCS, pp. 86–92 (2000)
23. Ravi, R.: Rapid rumor ramification: Approximating the minimum broadcast time. In: FOCS, pp. 202–213 (1994)
24. Ravi, R.: Matching Based Augmentations for Approximating Connectivity Problems. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 13–24. Springer, Heidelberg (2006)
25. Ravi, R., Goemans, M.X.: The constrained minimum spanning tree problem (extended abstract). In: Karlsson, R., Lingas, A. (eds.) SWAT 1996. LNCS, vol. 1097, pp. 66–75. Springer, Heidelberg (1996)
26. Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Hunt, H.B.: Many Birds with One Stone: Multi-objective Approximation Algorithms. In: STOC, pp. 438–447 (1993)
27. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Algorithms and Combinatorics, vol. 24. Springer, Berlin (2003)
28. Shmoys, D.B., Tardos, É.: Scheduling unrelated machines with costs. In: SODA, pp. 448–454 (1993)
29. Singh, M., Lau, L.C.: Approximating minimum bounded degree spanning trees to within one of optimal. In: STOC, pp. 661–670 (2007)